Distributed Systems

- **Textbook**

  Andrew S. Tanenbaum and Maarten van Steen, *Distributed Systems: Principles and Paradigms, 2nd edition*, Prentice Hall, 2007.

- **Reference**

  George Coulouris etc., *Distributed Systems: Concepts and Design, 4th edition*, Addison-Wesley, 2005.

Zhuzhong Qian

- Room 617, Computer Science Building
- Email: qzz@nju.edu.cn

- Research: distributed system, networking
  - Resource management for big data analytics
  - Network function virtualization
  - Mobile computing

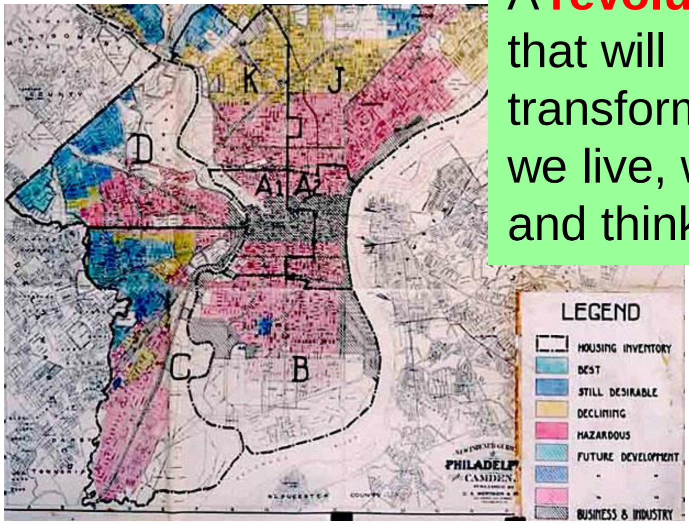
我们是否处于“第三次工业革命”的前夜？

或者说：
又一次“工业革命”已经开始！

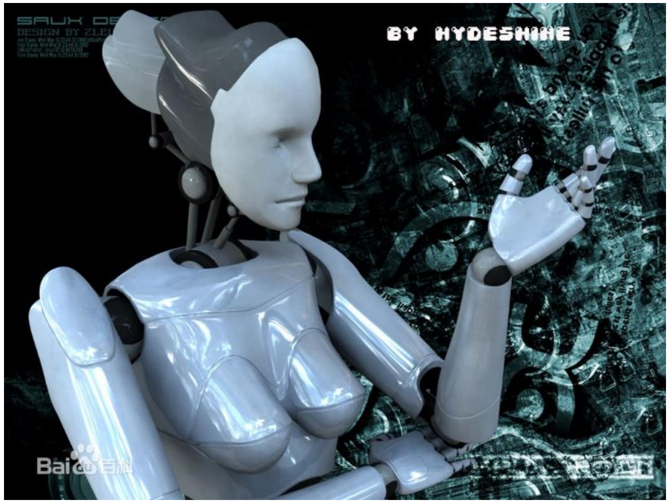
Everything Online

高度互联：anytime, anywhere, any way!

大数据：
A **revolution** that will transform how we live, work, and think!



人工智能：
Simulation and extension of human intelligence.



什么是背后的推动力?

计算机科学在新一轮变革中必将承担以前数学与自然科学所承担的使命。

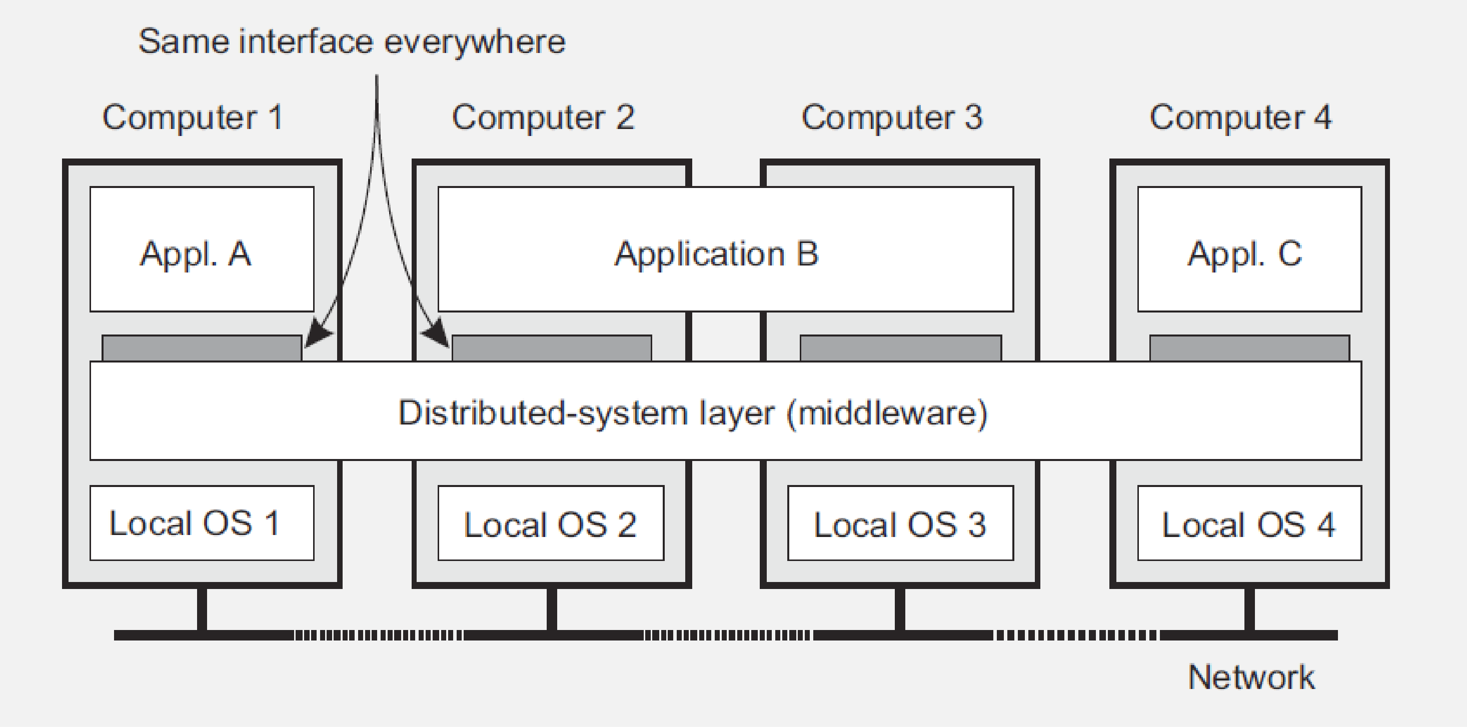
分布式系统是构成当代信息系统的重要基础架构。

Introduction

Distributed Systems [1]

Distributed Systems: Definition

- A distributed system is a collection of *autonomous computing elements* that appears to its users as a *single coherent system*.



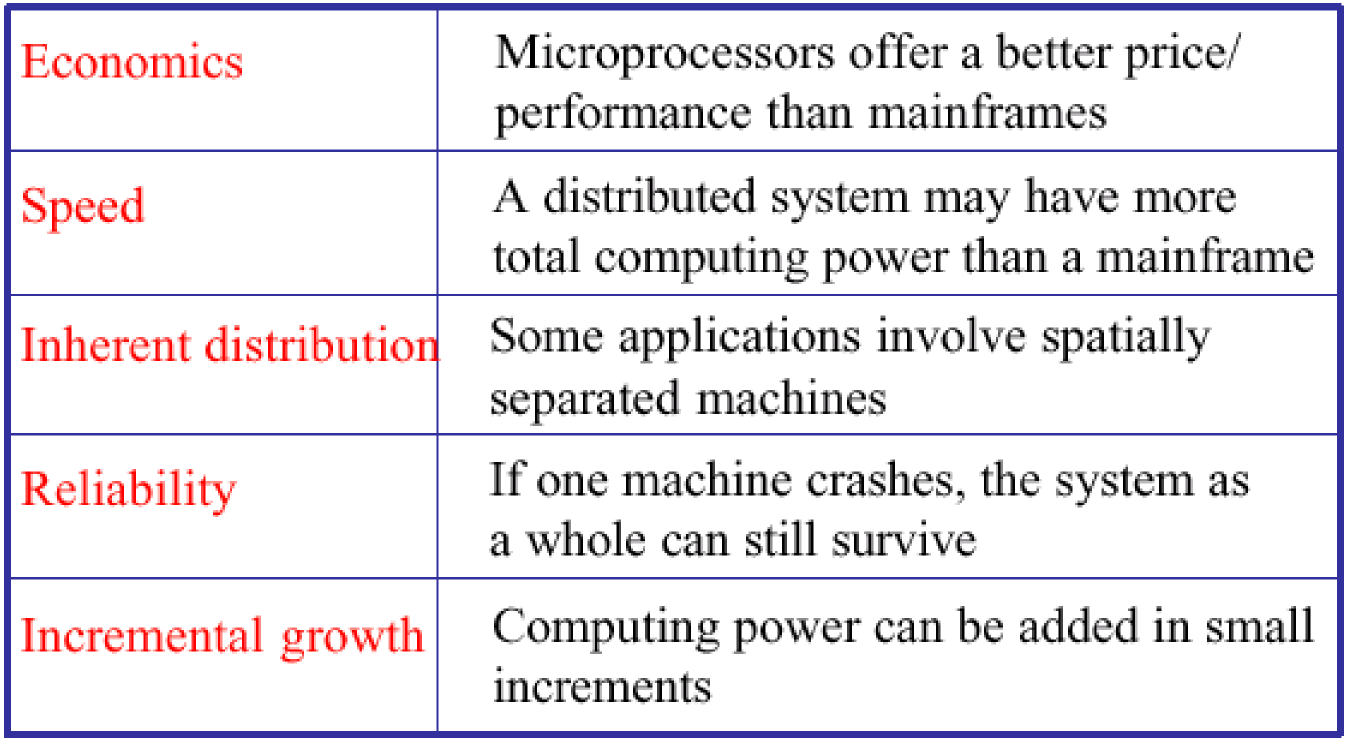
Examples

- A network of workstations allocated to users
- A pool of processors in the machine room allocated dynamically
- A single file system (all users access files with the same path name)
- User command executed in the best place (user workstation, a workstation belonging to someone else, or on an unassigned processor in the machine room)

Technology advances

- Networking
- Processors
- Memory
- Storage
- Protocol

Why Distributed?



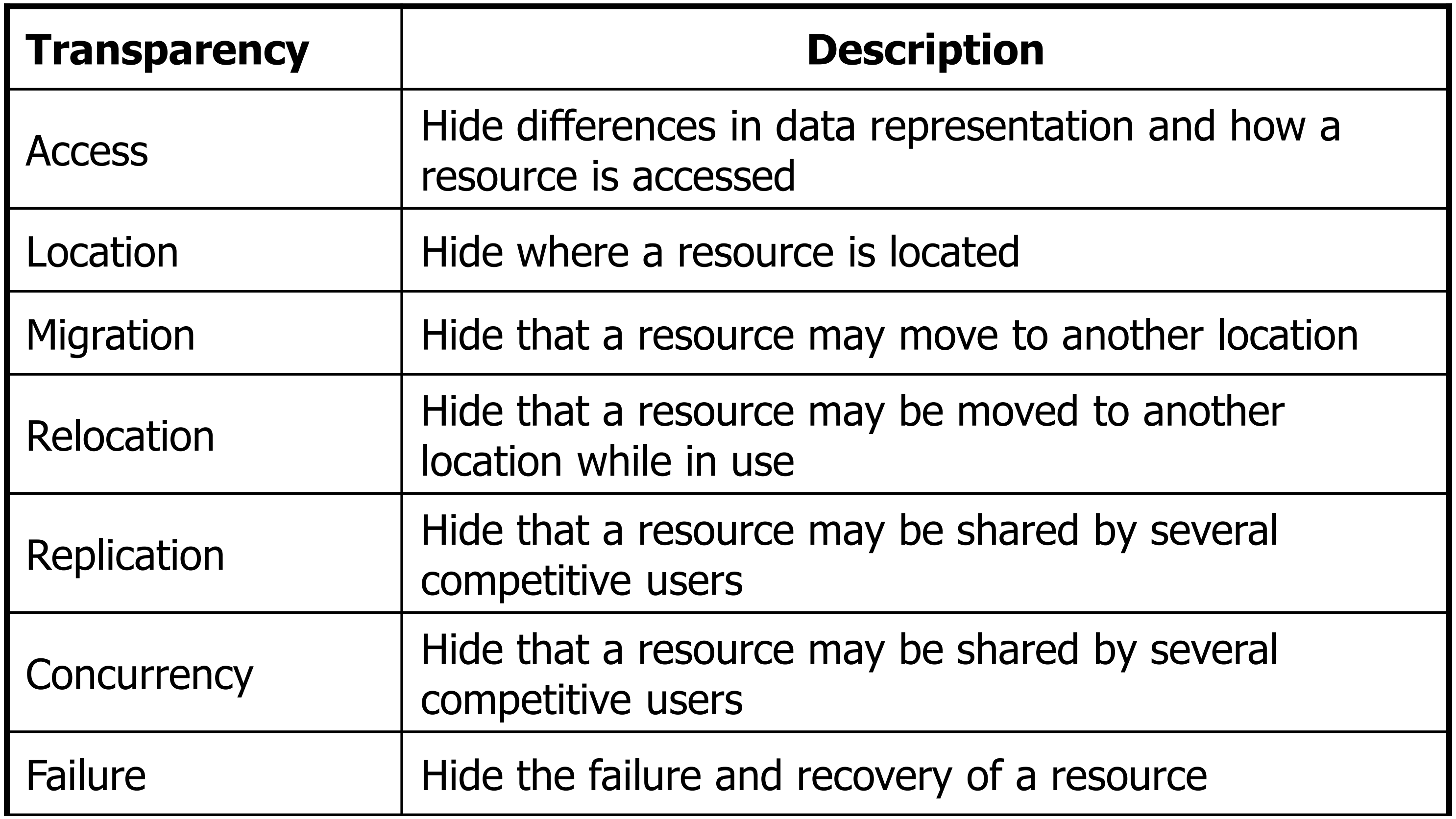
Economics	Microprocessors offer a better price/ performance than mainframes
Speed	A distributed system may have more total computing power than a mainframe
Inherent distribution	Some applications involve spatially separated machines
Reliability	If one machine crashes, the system as a whole can still survive
Incremental growth	Computing power can be added in small increments

People are distributed

Goals of Distributed Systems

- Making resources available
- Distribution transparency
- Openness
- Scalability

Transparency in a Distributed System



Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource may be shared by several competitive users
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource

Distribution transparency is a nice goal, but achieving it is a different story.

Degree of Transparency

- Aiming at full distribution transparency may be too much:
  - Users may be located in different continents
  - Completely hiding failures of networks and nodes is (theoretically and practically) impossible
    - You cannot distinguish a slow computer from a failing one
    - You can never be sure that a server actually performed an operation before a crash
  - Full transparency will cost performance, exposing distribution of the system
    - Keeping Web caches exactly up-to-date with the master
    - Immediately flushing write operations to disk for fault tolerance

Openness of Distributed Systems

- Be able to interact with services from other open systems, irrespective of the underlying environment:
  - Systems should conform to well-defined interfaces
  - Systems should support portability of applications
  - Systems should easily interoperate

- At least make the distributed system independent from heterogeneity of the underlying environment.

Policies Versus Mechanisms

- Requires support for different policies:
  - What level of consistency do we require for client-cached data?
  - Which operations do we allow downloaded code to perform?
  - Which QoS requirements do we adjust in the face of varying bandwidth?
  - What level of secrecy do we require for communication?
- Ideally, a distributed system provides only mechanisms:
  - Allow (dynamic) setting of caching policies
  - Support different levels of trust for mobile code
  - Provide adjustable QoS parameters per data stream
  - Offer different encryption algorithms

Scalability

- At least three components:
  - Number of users and/or processes (size scalability)
  - Maximum distance between nodes (geographical scalability)
  - Number of administrative domains (administrative scalability)

- Most systems account only, to a certain extent, for size scalability. The (non)solution: powerful servers. Today, the challenge lies in **geographical** and **administrative** scalability.

Types of Distributed Systems

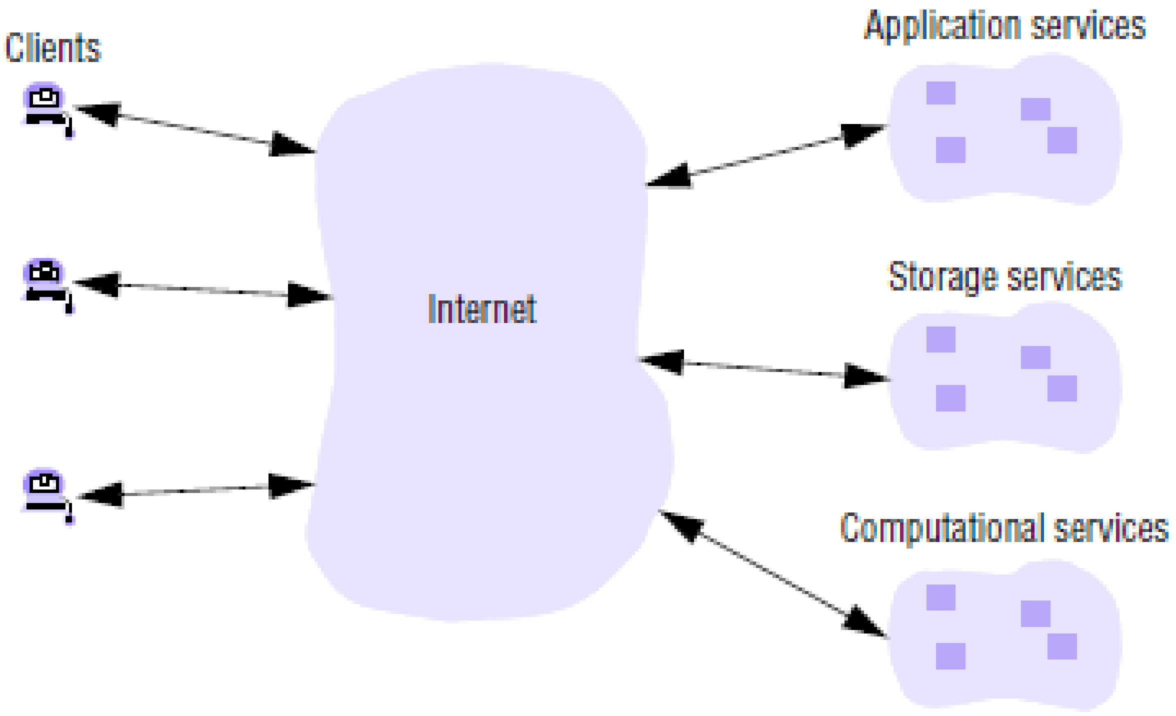
- Distributed computing systems
- Distributed information systems
- Distributed pervasive systems

Distributed Computing Systems

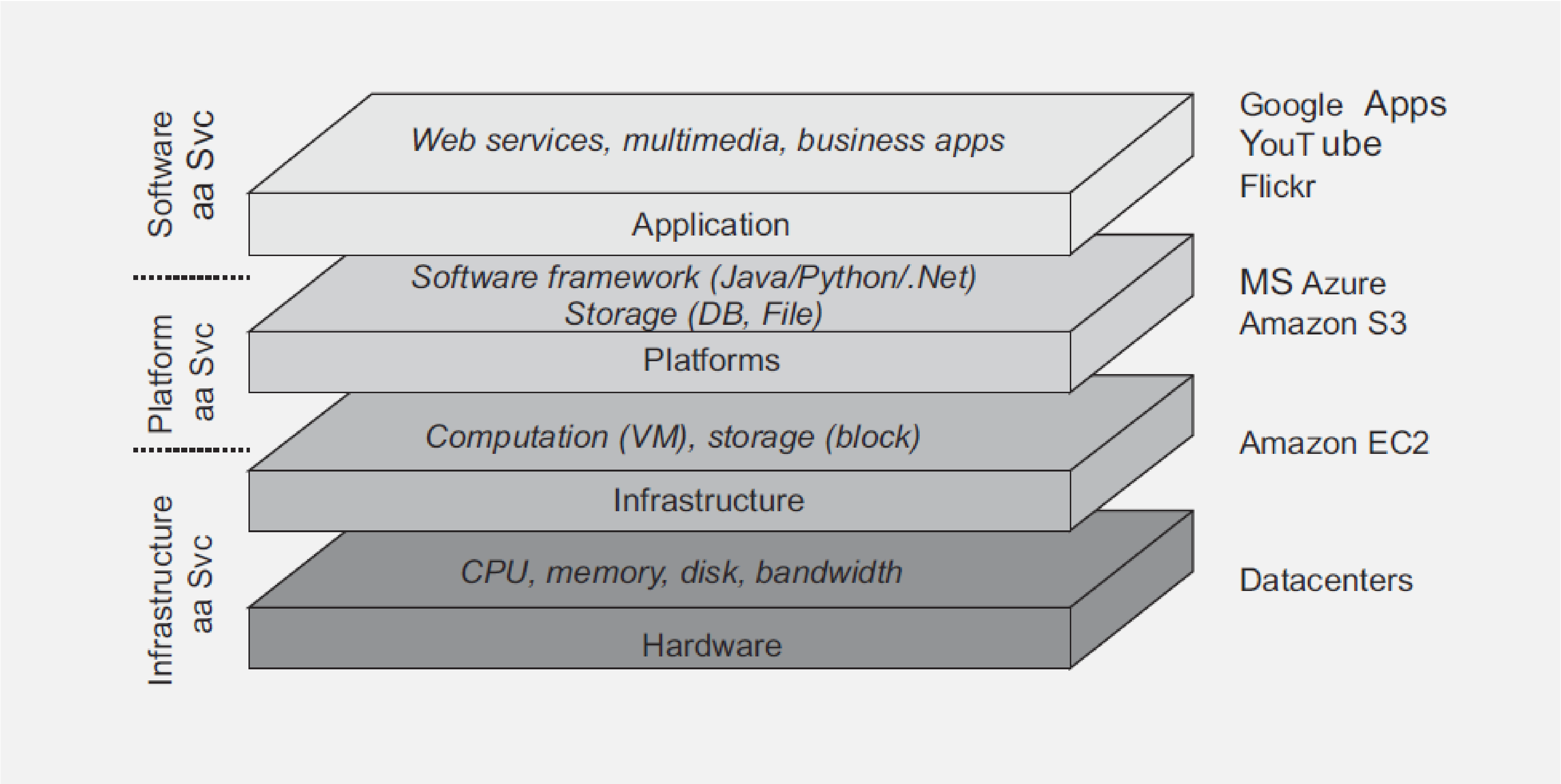
- **Cluster Computing**
  - Essentially a group of high-end systems connected through a LAN:
    - Homogeneous: same OS, near-identical hardware
    - Single managing node

- **Grid Computing**
  - The next step: lots of nodes from everywhere:
    - Heterogeneous
    - Dispersed across several organizations
    - Can easily span a wide-area network

Distributed Computing Systems: Clouds



Cloud Platform

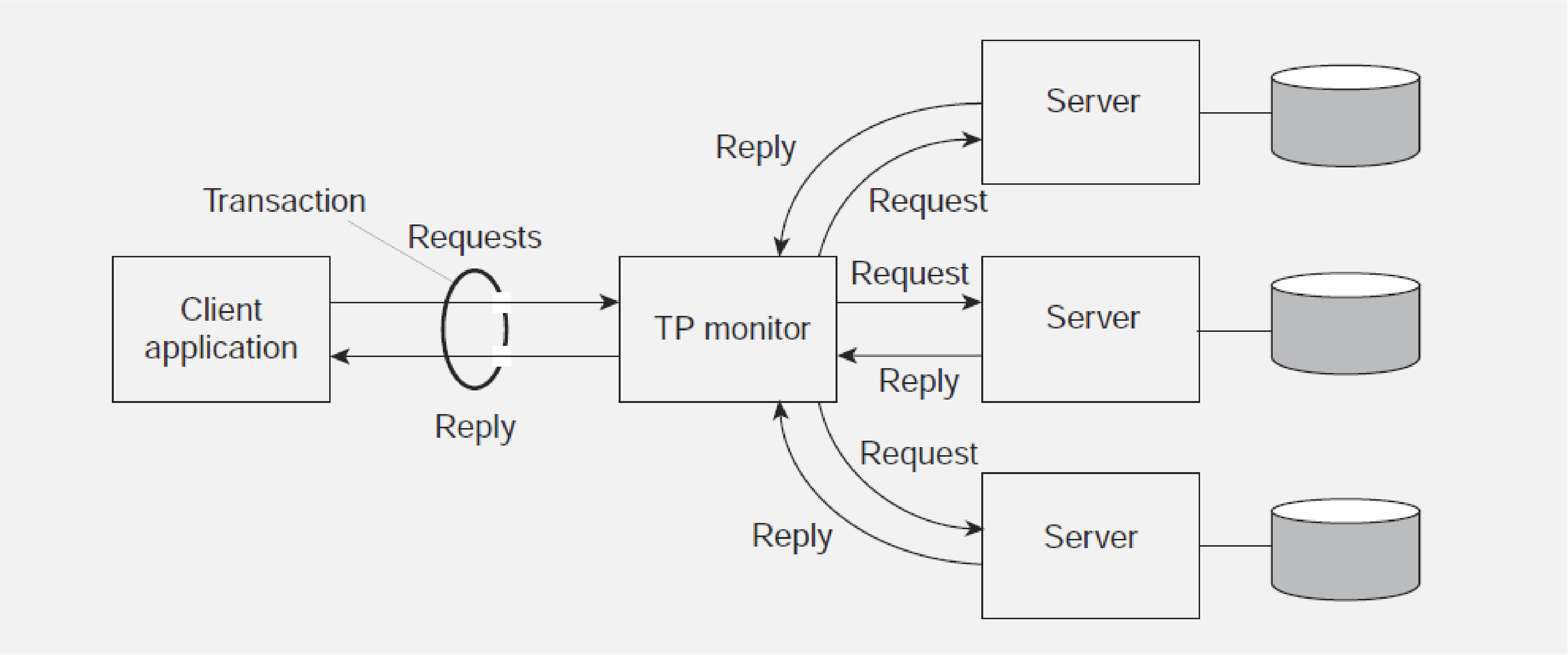


Distributed Information Systems

- The vast amount of distributed systems in use today are forms of traditional information systems, that now integrate legacy systems.
  - Example: Transaction processing systems.
- A transaction is a collection of operations on the state of an object (database, object composition, etc.) that satisfies the following properties (ACID)
  - Atomicity
  - Consistency
  - Isolation
  - Durability

Transaction Processing Monitor

- In many cases, the data involved in a transaction is distributed across several servers. A TP Monitor is responsible for coordinating the execution of a transaction.



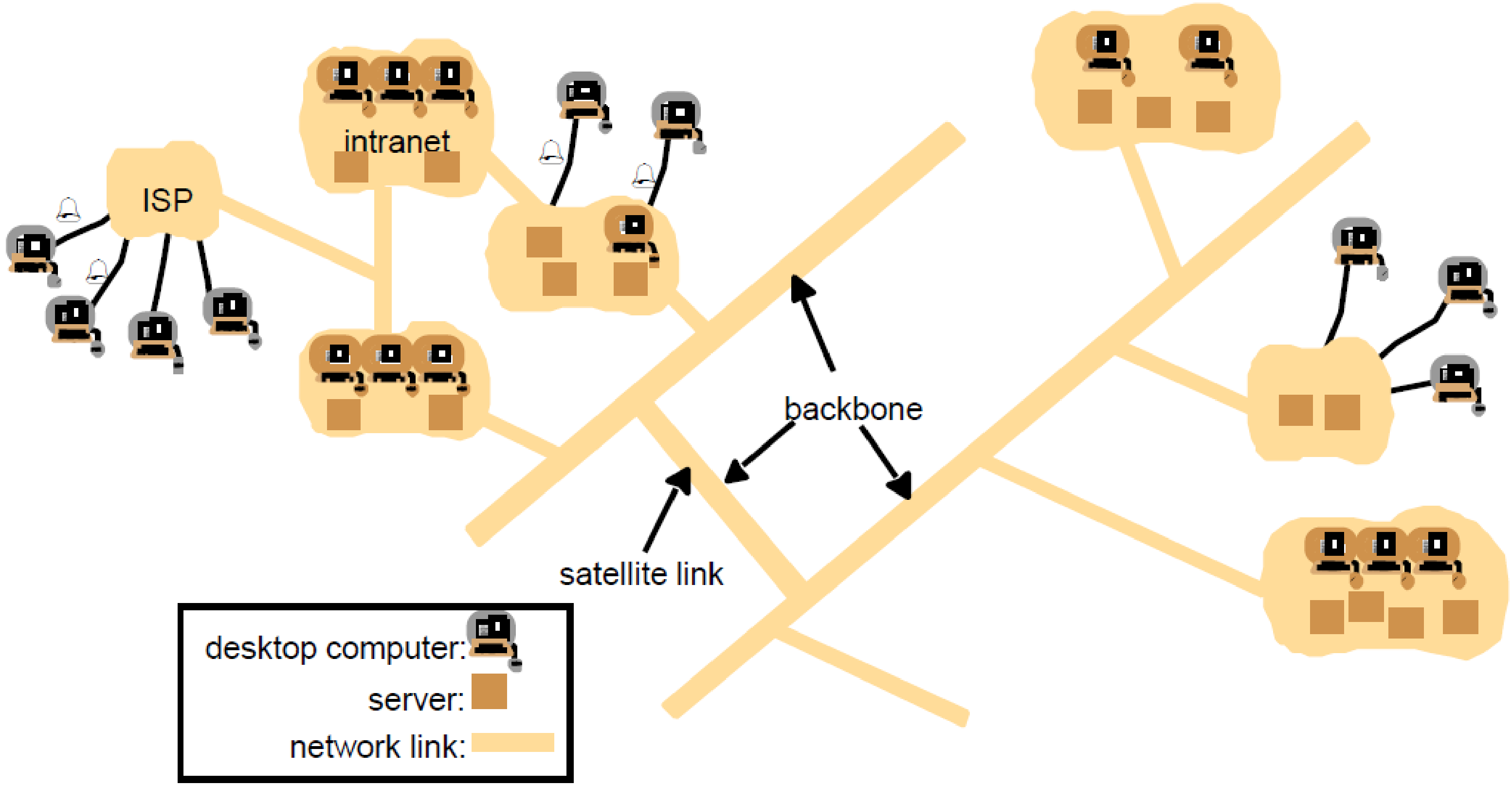
Distributed Pervasive Systems

- Emerging next-generation of distributed systems in which nodes are small, mobile, and often embedded in a larger system, characterized by the fact that the system naturally blends into the user’s environment.
  - Ubiquitous computing systems
  - Mobile computing systems
  - Sensor (and actuator) networks

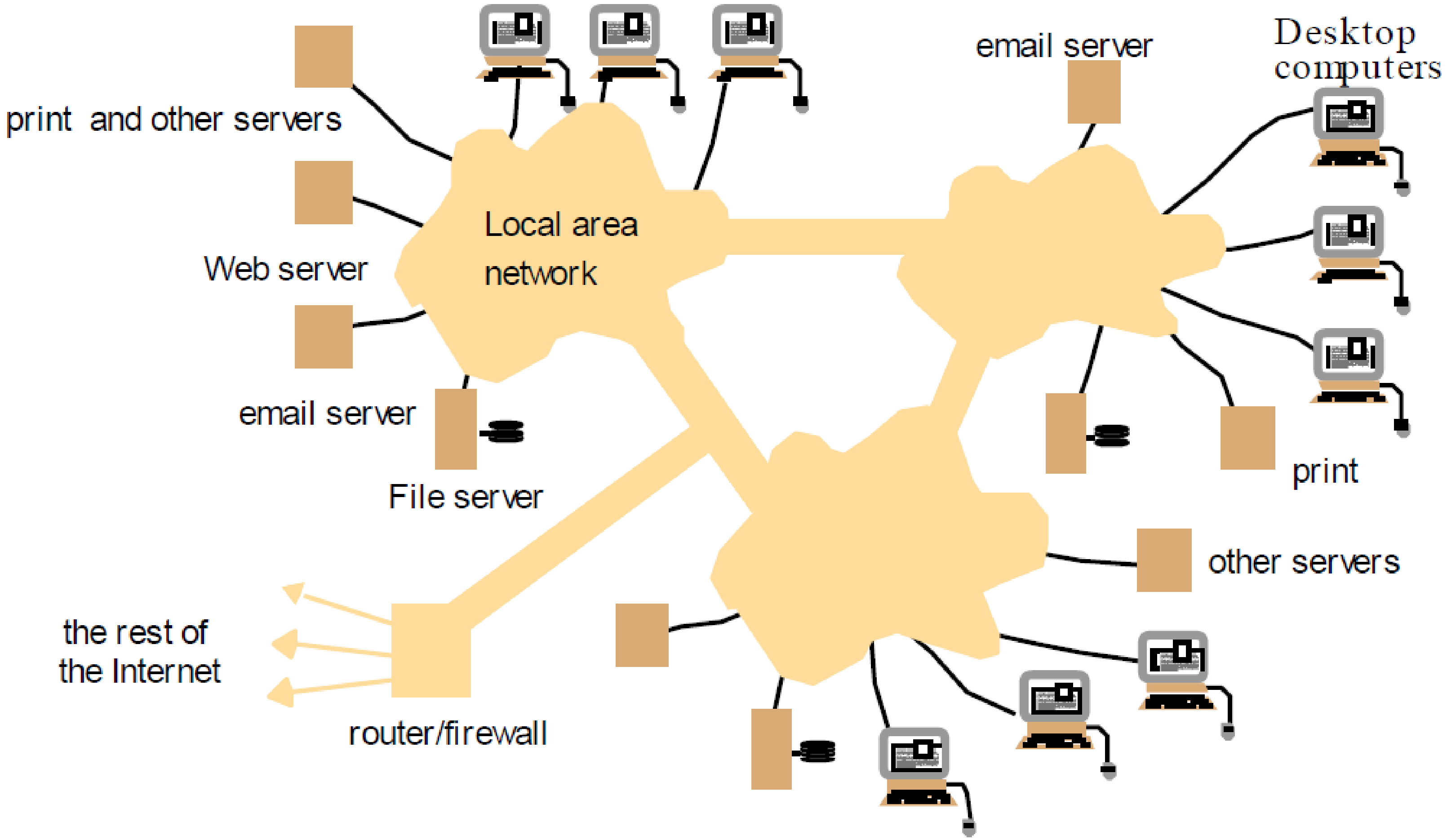
Some Typical Examples

- Internet
- Intranet
- Mobile environment
- Web

Internet

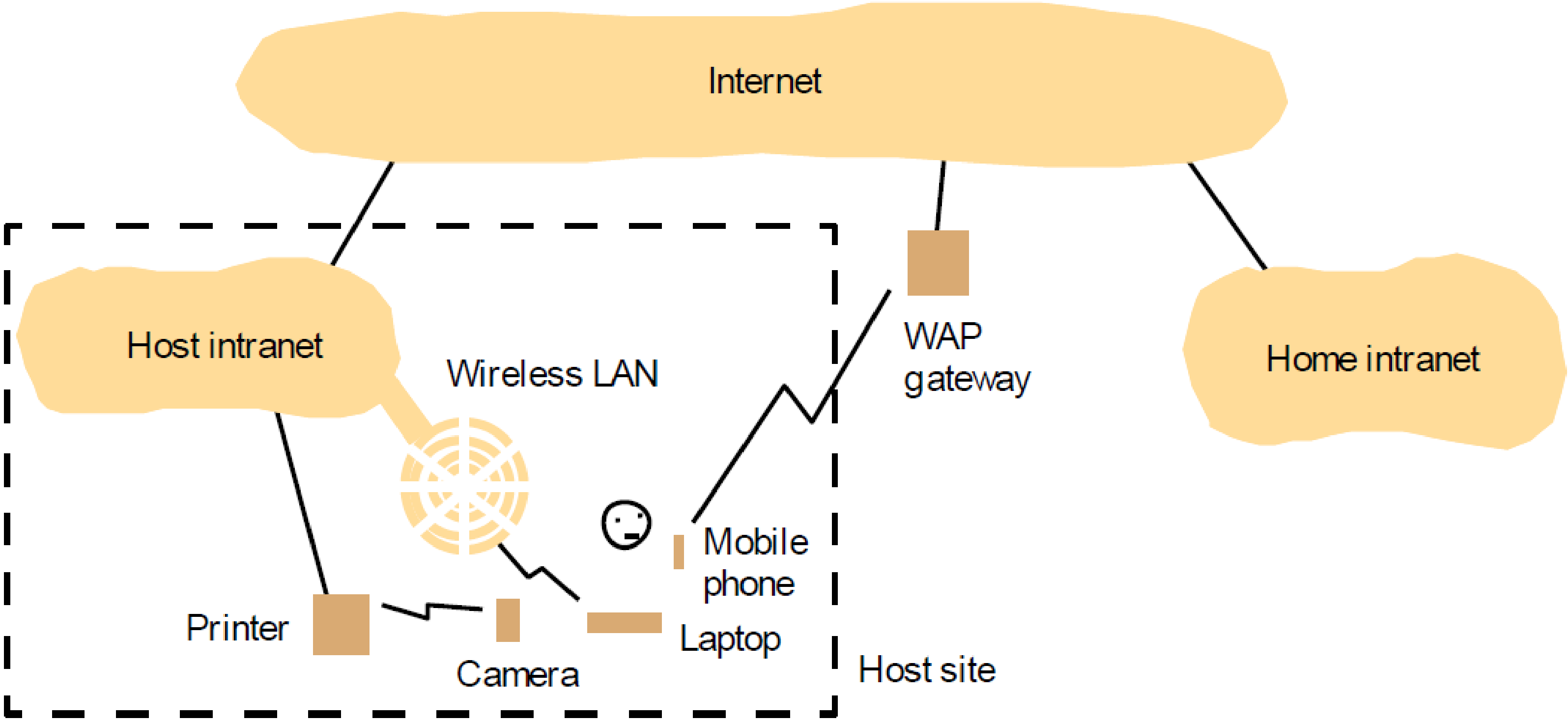

intranet
ISP
backbone
satellite link
desktop computer:
server:
network link:

Intranet

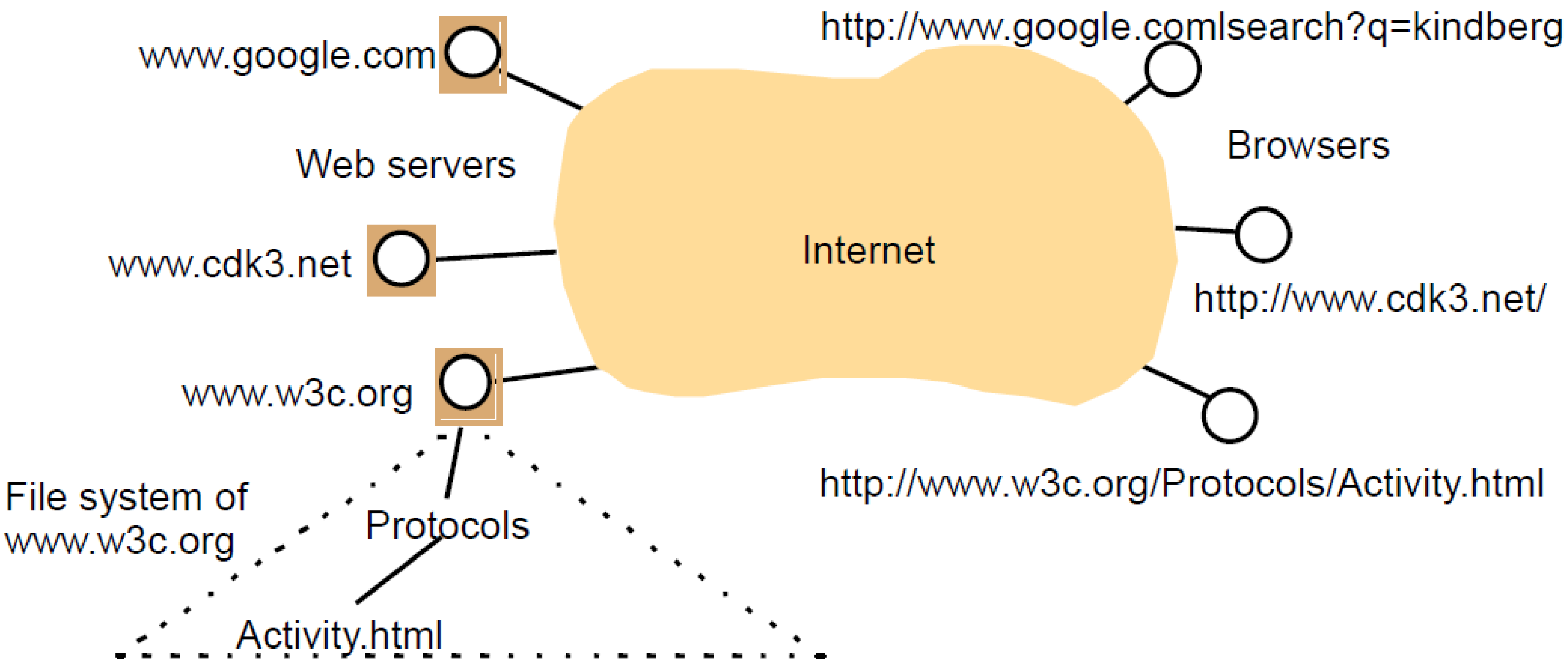


print and other servers
Web server
email server
File server
Local area
network
email server
Desktop
computers
print
other servers
the rest of
the Internet
router/firewall

Mobile Environment



Web


www.google.com
http://www.google.comlsearch?q=kindberg
Web servers
Browsers
www.cdk3.net
Internet
http://www.cdk3.net/
www.w3c.org
File system of
www.w3c.org
Protocols
http://www.w3c.org/Protocols/Activity.html
Activity.html